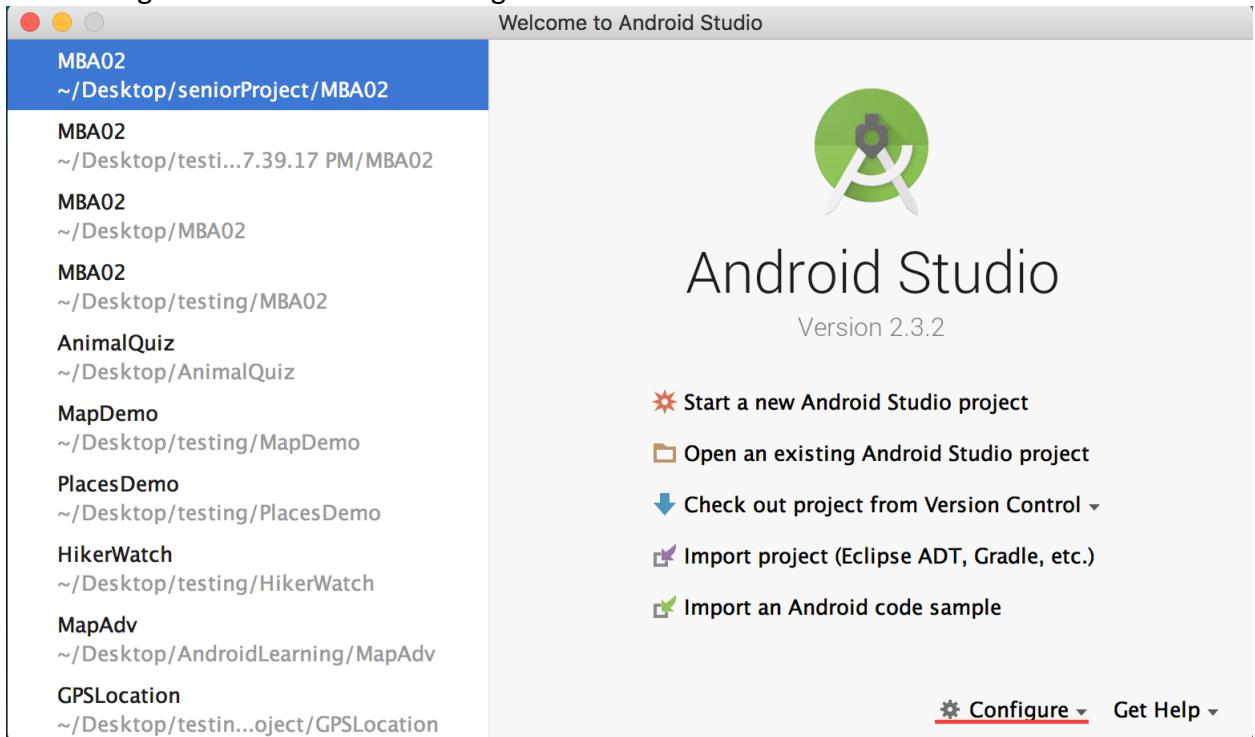


## CSE575-School-App

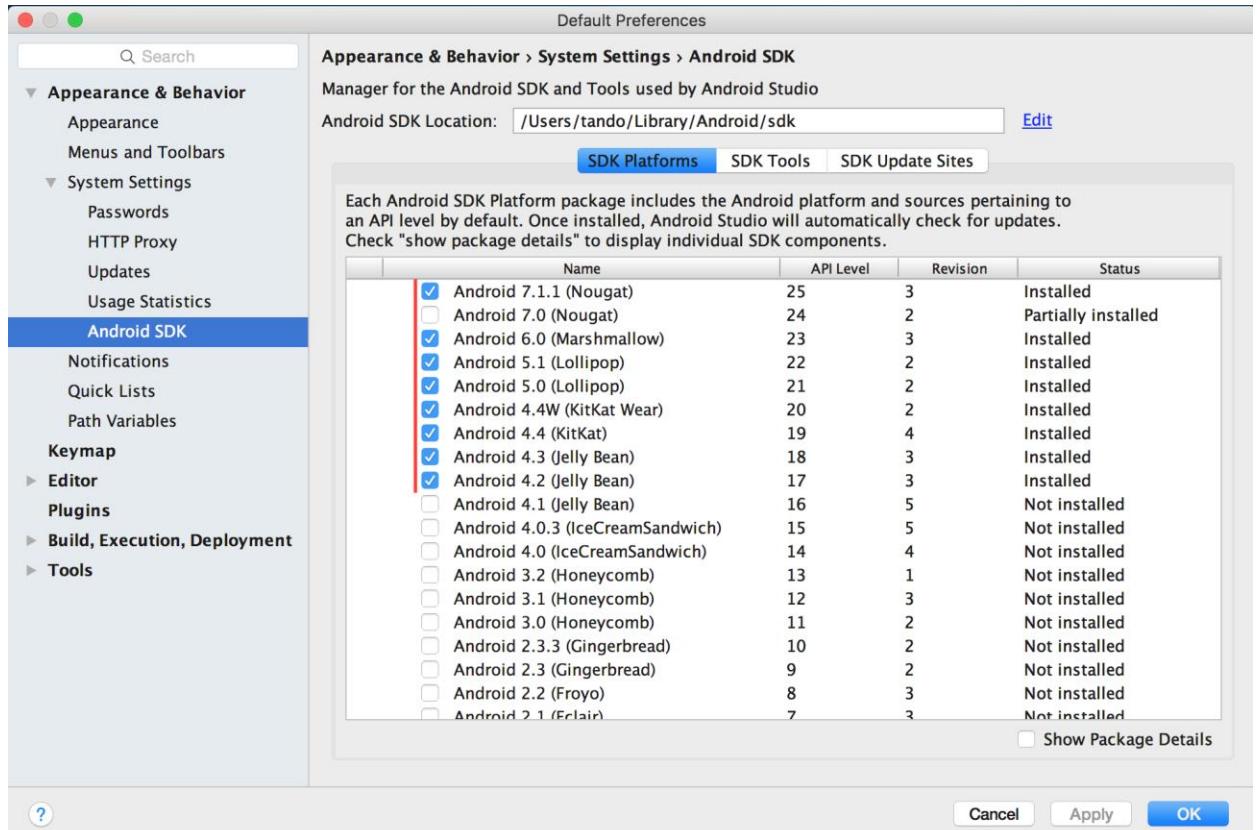
### Step by Step tutorial

#### I. Preparation Android Studio

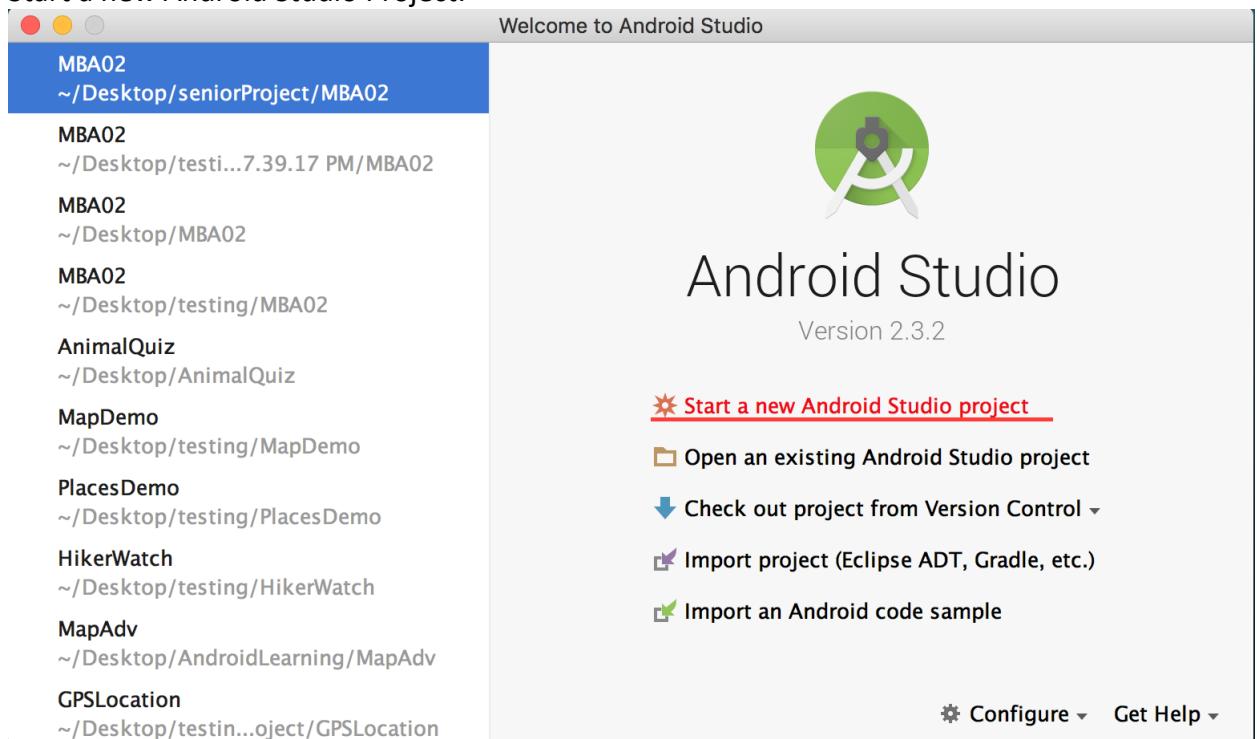
1. Download (Link to download: <https://developer.android.com/studio/index.html>) and Open Android Studio.
2. Click configure and choose SDK manager



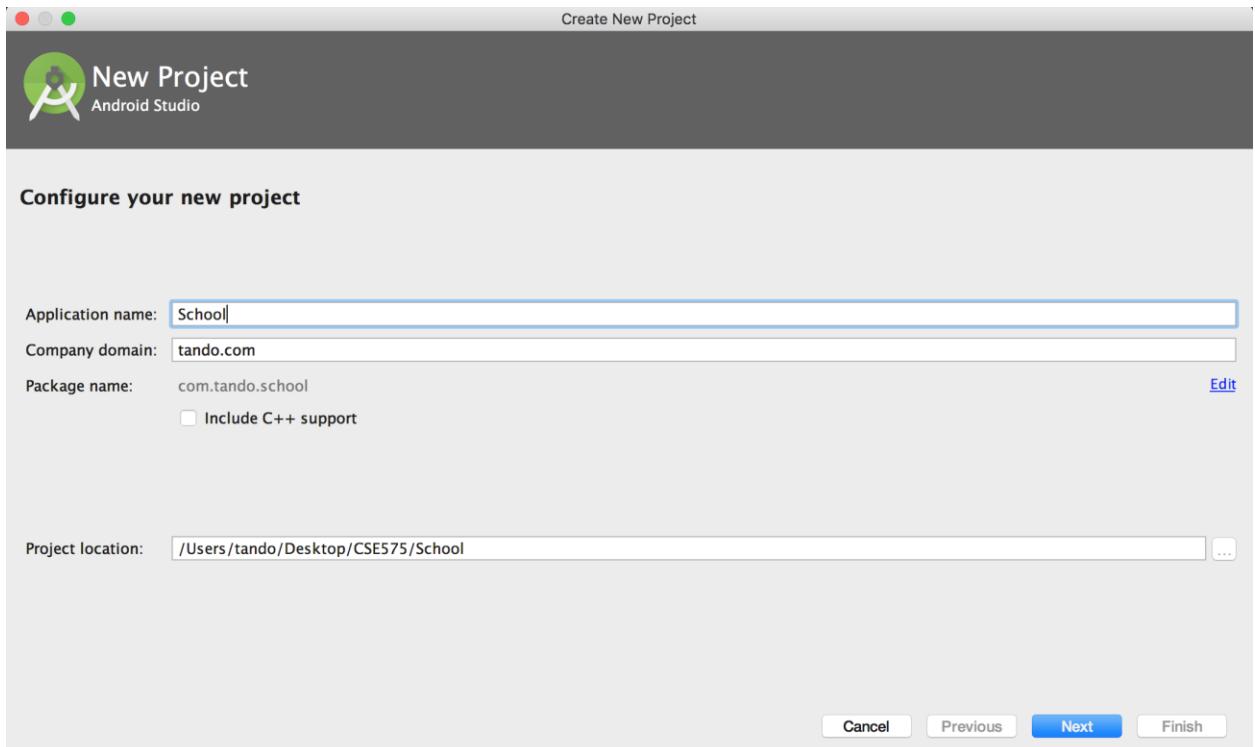
3. Install all the SDK platforms that you need for your project. In this case, I picked the minimum SDK is 4.2 to the latest one.



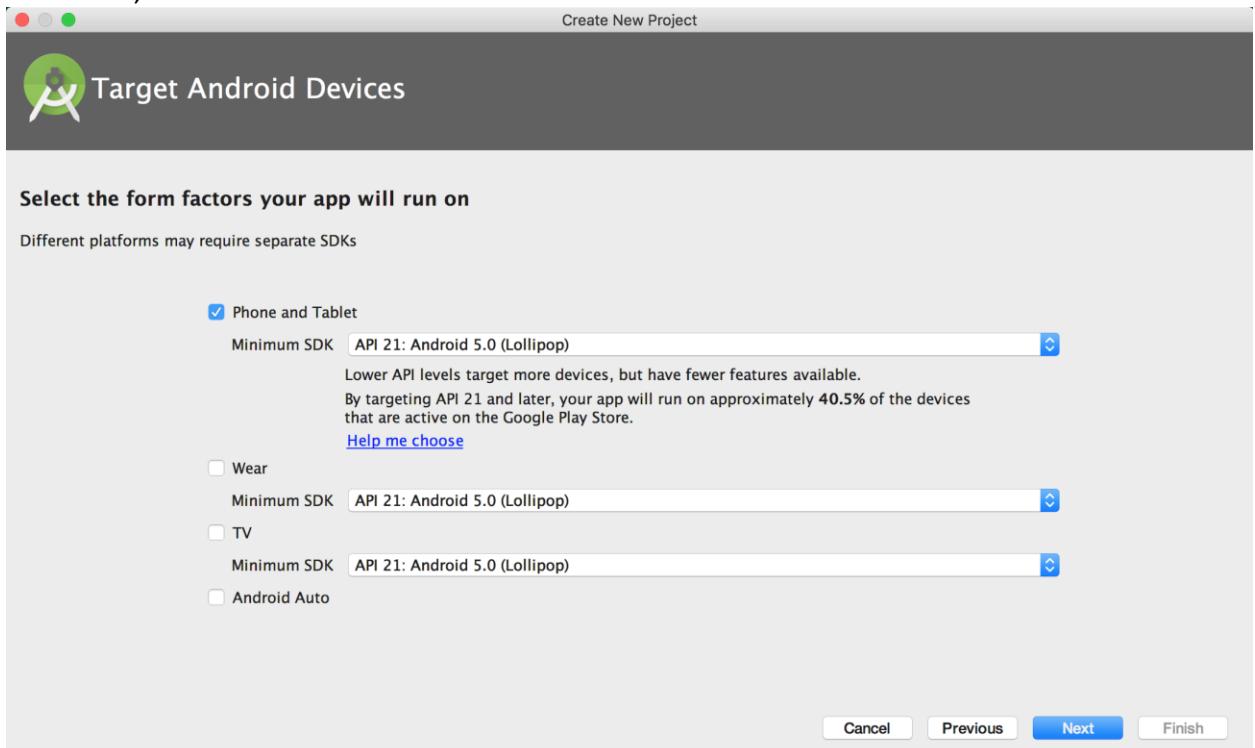
- After choosing the SDK platforms, click ok and wait for them to be installed. Then, click Start a new Android Studio Project.



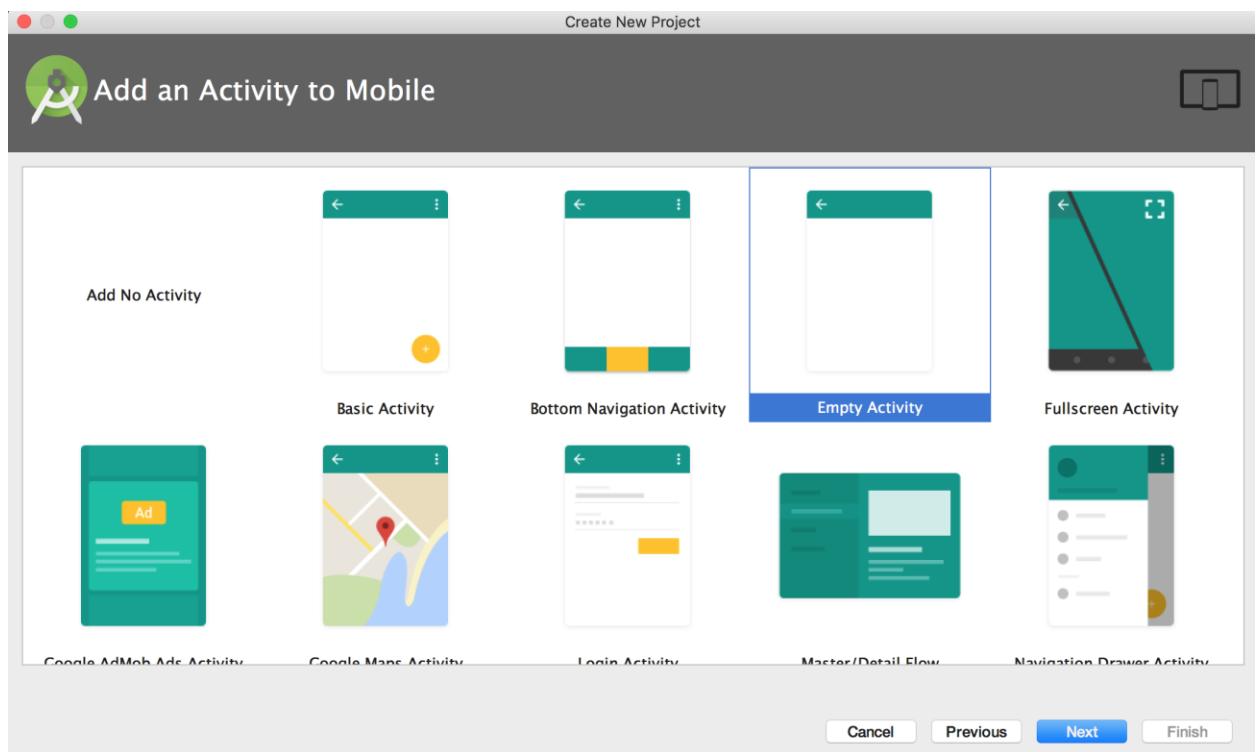
5. Name your project, in this case I chose my app name as School. In the Project location section, choose the directory that you want your app to be located.



6. Click Next, and choose the minimum SDK

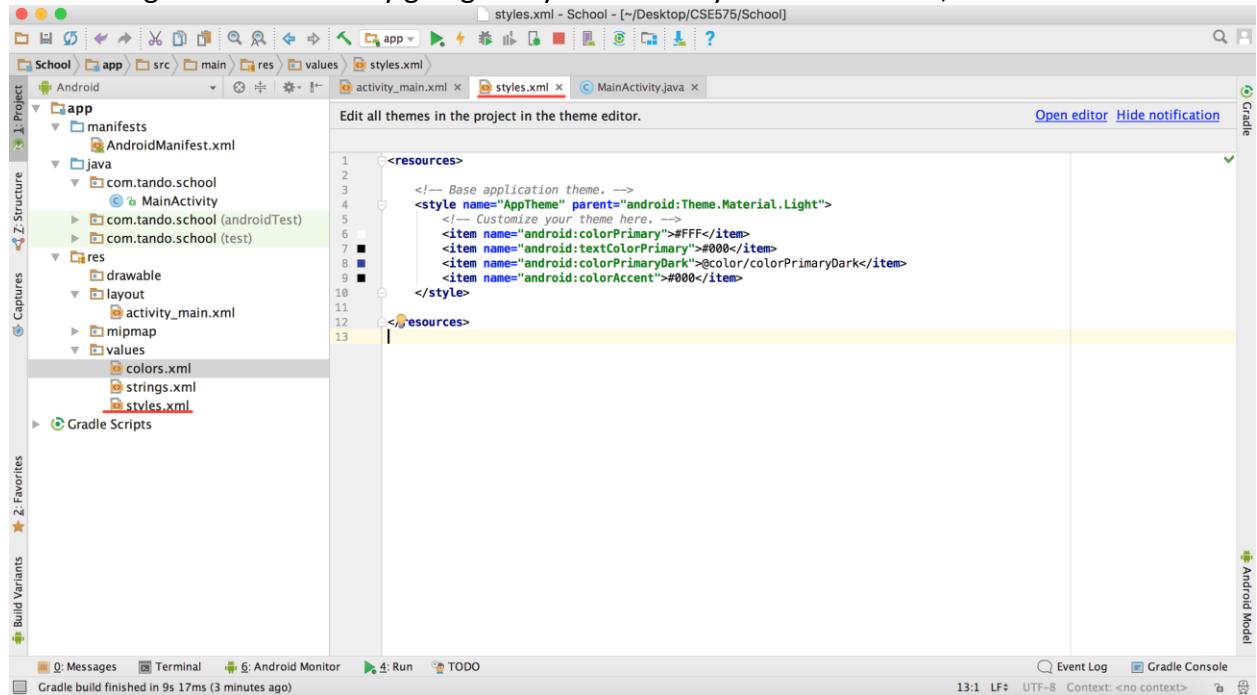


7. Choose a Empty Activity, click Next, and click Finish

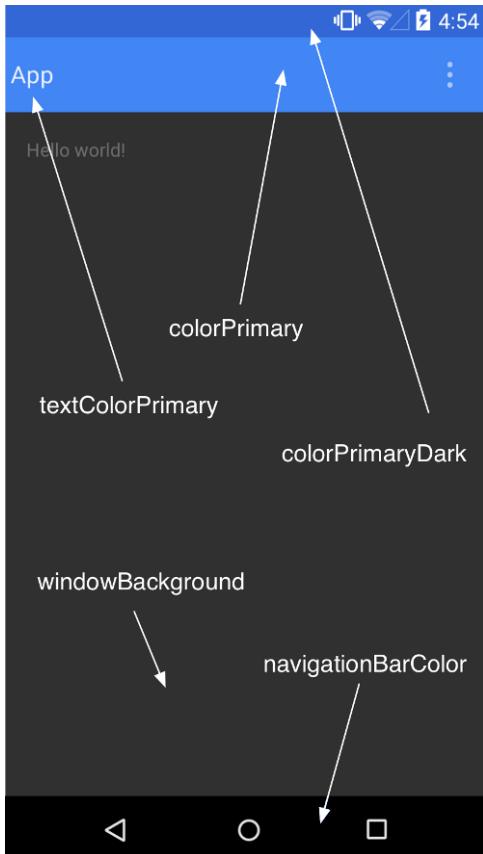


## II. Working with the front page (main activity) of the app

### 1. Starting to make the UI by going to styles.xml to style the Action bar, and text color



You can style any color that you like by following this:



Reference: <https://developer.android.com/training/material/theme.html>

Here is the customized styled.xml for the app:

```
<resources xmlns:android="http://schemas.android.com/tools">

    <!-- Base application theme. -->
    <style name="MyTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
        <item name="android:colorPrimary">#FFF</item>
        <item name="android:colorPrimaryDark">@color/colorPrimaryDark</item>
        <item name="android:colorAccent">#000</item>
        <item name="android: actionBarStyle">@style/MyActionBar</item>
        <item name="actionBarStyle">@style/MyActionBar</item>
    </style>

    <!-- Customized ActionBar. Change its background to white color -->
    <style name="MyActionBar" parent="@style/Widget.AppCompat.Light.ActionBar">
        <item name="android:background">#FFF</item>
        <item name="android:titleTextStyle">@style/MyActionBarTitleText</item>
        <item name="titleTextStyle">@style/MyActionBarTitleText</item>
    </style>

    <!-- Customized text title on the ActionBar. Change to black color. -->
    <style name="MyActionBarTitleText" parent="@style/TextAppearance.AppCompat.Widget.ActionBar.Title">
        <item name="android:textColor">#000</item>
    </style>

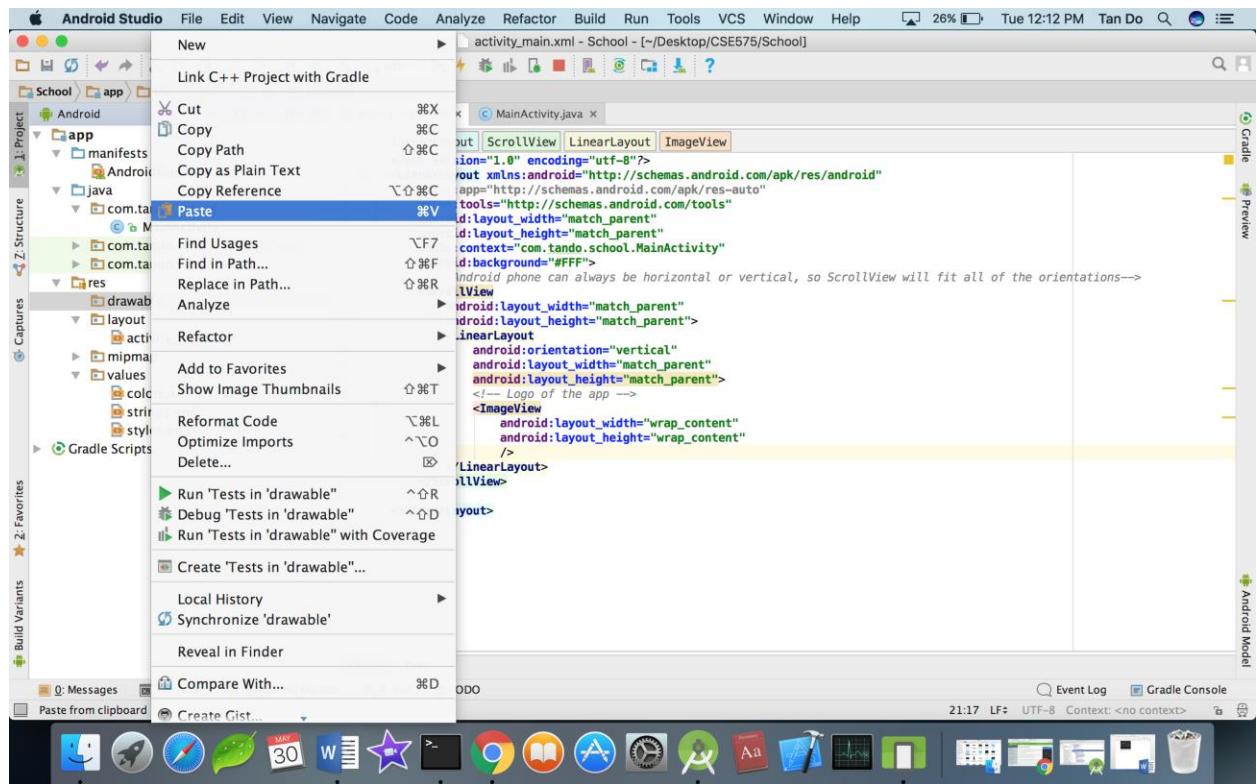
```

```
</style>  
</resources>
```

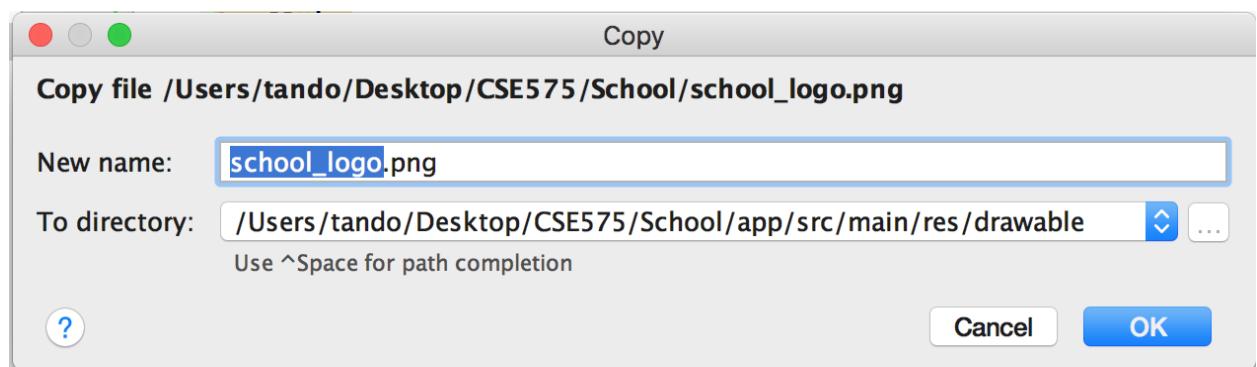
## 2. Create the UI for the main page:

Import images to the drawable directory, so we can have images logos, backgrounds, icons, and so much more

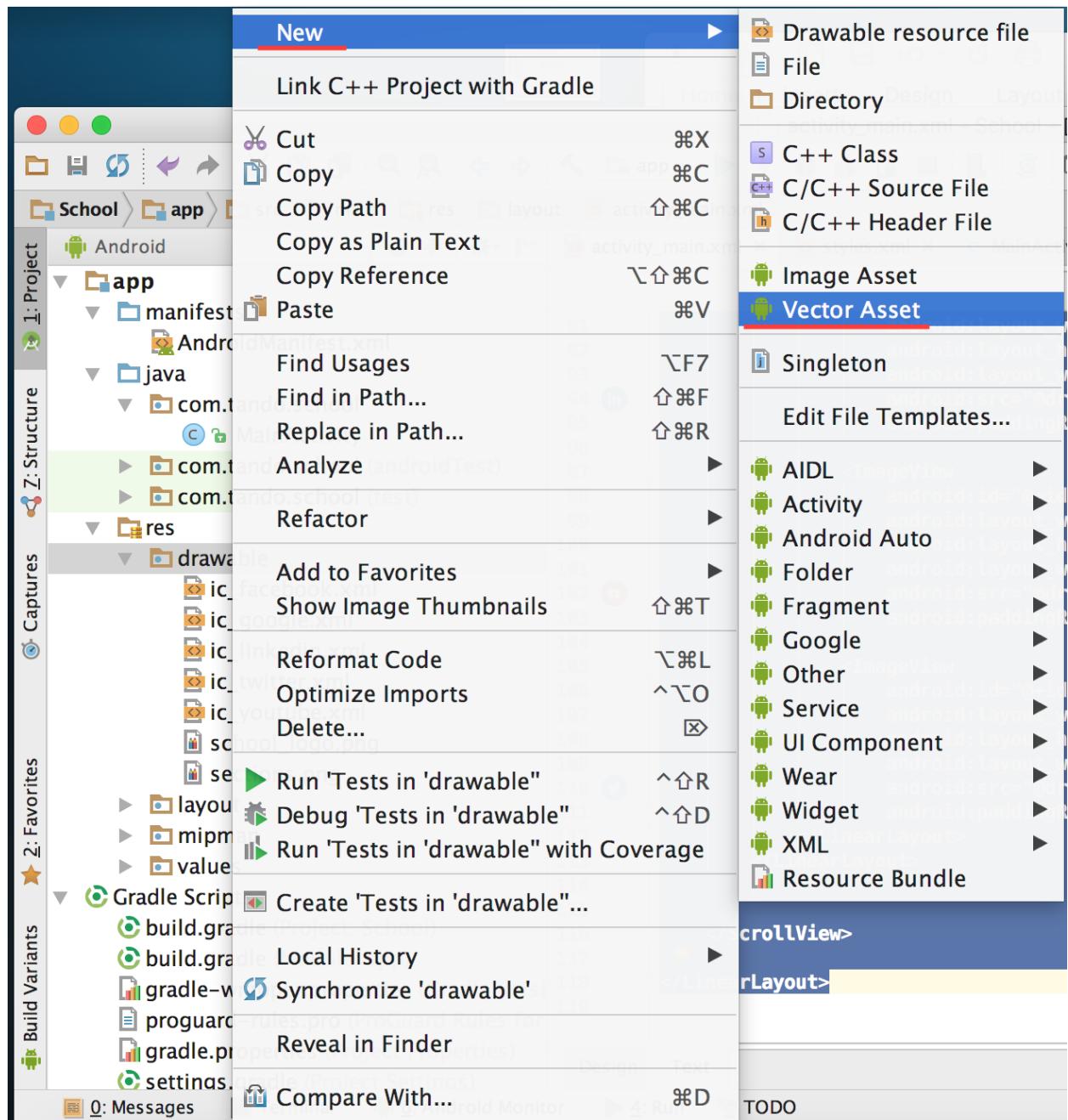
Copy the image (which should be saved in your local directory) that you want to import. Right click on drawable directory and click paste.

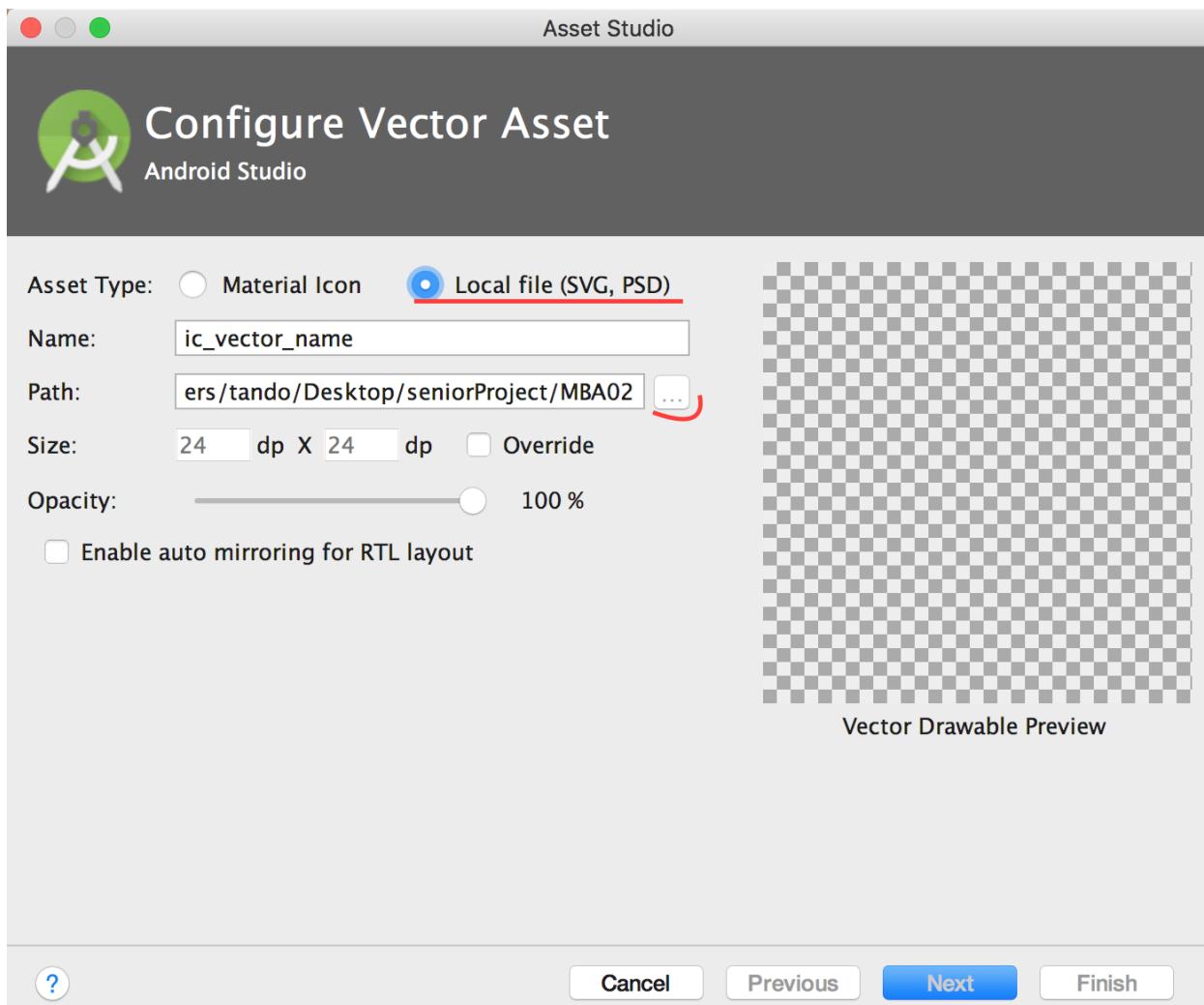


Then click OK



There will be some images that are in SVG type, then you need to right click on drawable directory > click new > Vector Asset > click Local file > go to the directory that you have saved your images > NEXT > FINISH.





Create an UI could be done by 2 methods, Design and Text. I used Text in this case.

So my main\_activity.xml look like this:

Notice that there are onClick method for each buttons and icons, this method will be called later to make buttons and images functioning such as clickable. The method's names should be unique and meaningful.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.tando.school.MainActivity"
    android:background="#FFF"
    android:orientation="vertical">
    <!-- Android phone can always be horizontal or vertical, so ScrollView will fit all of the orientations-->
    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent">
        <LinearLayout
            android:orientation="vertical"
```

```
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginBottom="10dp">

    <!--Logo -->
    <ImageView
        android:id="@+id/logo"
        android:layout_width="150dp"
        android:layout_height="100dp"
        android:background="@drawable/school_logo"
        android:layout_gravity="center"/>

    <!--4 sections of the app -->
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="My School Account"
        android:textColor="#FFFFFF"
        android:layout_marginBottom="20dp"
        android:layout_marginTop="10dp"
        android:background="@drawable/sections"
        android:onClick="StartAccount"
        />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Announcement"
        android:textColor="#FFFFFF"
        android:layout_marginBottom="20dp"
        android:background="@drawable/sections"
        android:onClick="StartAnnoucement"
        />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Feedback/Advising"
        android:textColor="#FFFFFF"
        android:layout_marginBottom="20dp"
        android:background="@drawable/sections"
        android:onClick="StartFeedback"
        />
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Utilities"
        android:textColor="#FFFFFF"
        android:layout_marginBottom="20dp"
        android:background="@drawable/sections"
        android:onClick="StartUtils"
        />
    <!--Social Media -->
    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
    android:layout_marginTop="10dp"
    android:orientation="horizontal"
    android:layout_gravity="center">

    <ImageView
        android:id="@+id/facebook"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:src="@drawable/ic_facebook"
        android:paddingRight="5dp"
        android:onClick="fbClick"/>

    <ImageView
        android:id="@+id/googlePlus"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:src="@drawable/ic_google"
        android:paddingRight="5dp"
        android:onClick="googleClick"/>

    <ImageView
        android:id="@+id/linkedin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:src="@drawable/ic_linkedin"
        android:paddingRight="5dp"
        android:onClick="linkedinClick"/>

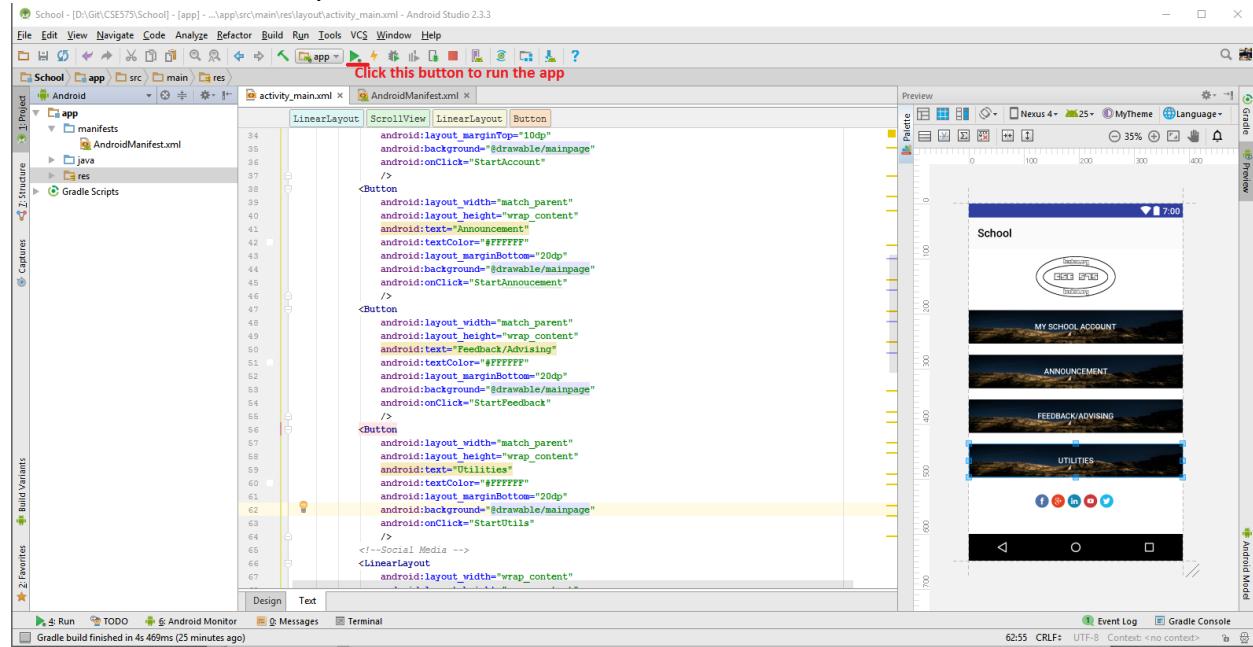
    <ImageView
        android:id="@+id/youtube"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:src="@drawable/ic_youtube"
        android:paddingRight="5dp"
        android:onClick="youTubeClick"/>

    <ImageView
        android:id="@+id/twitterT"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:src="@drawable/ic_twitter"
        android:paddingRight="5dp"
        android:onClick="twitterClick"/>
</LinearLayout>
</LinearLayout>

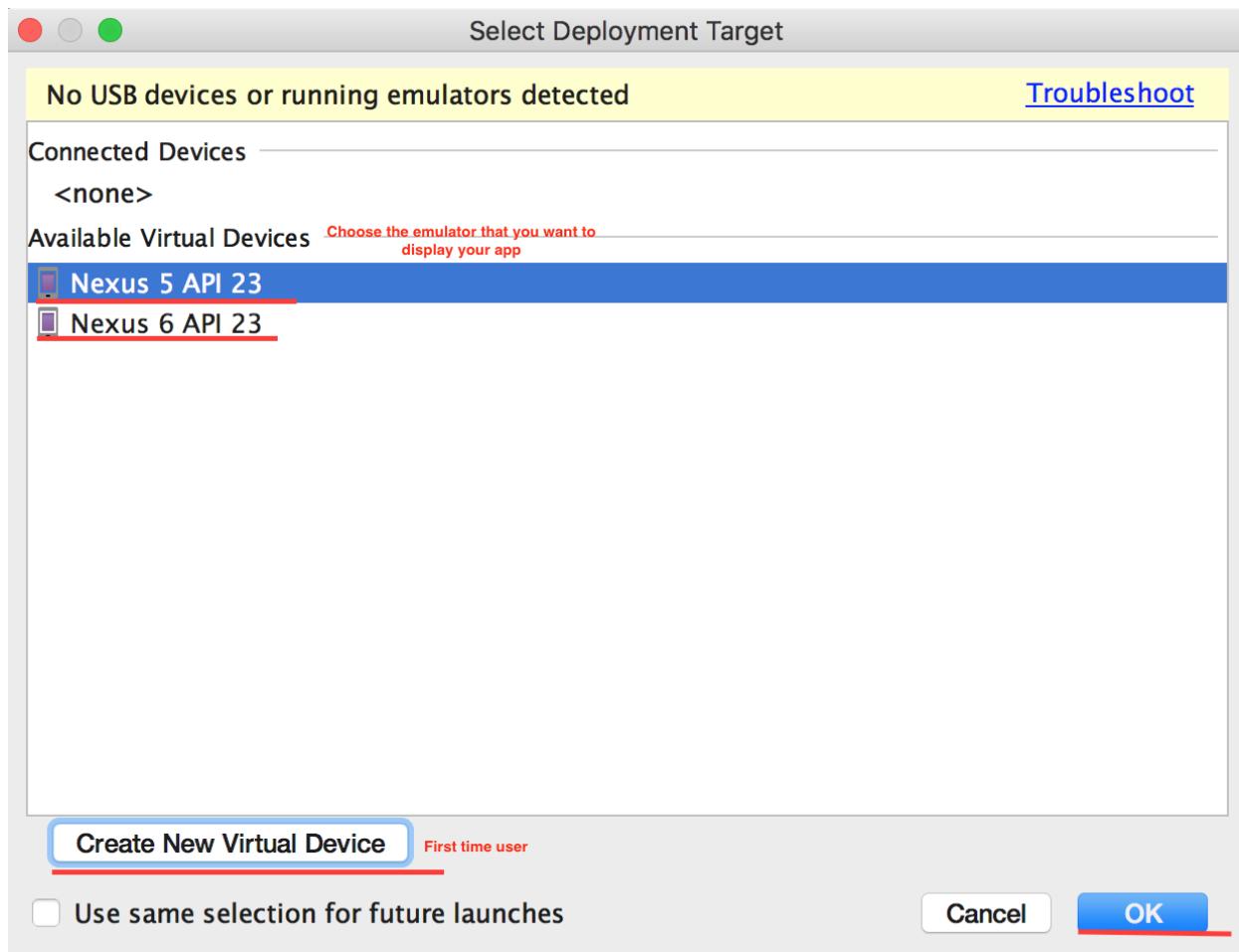
</ScrollView>
```

</LinearLayout>

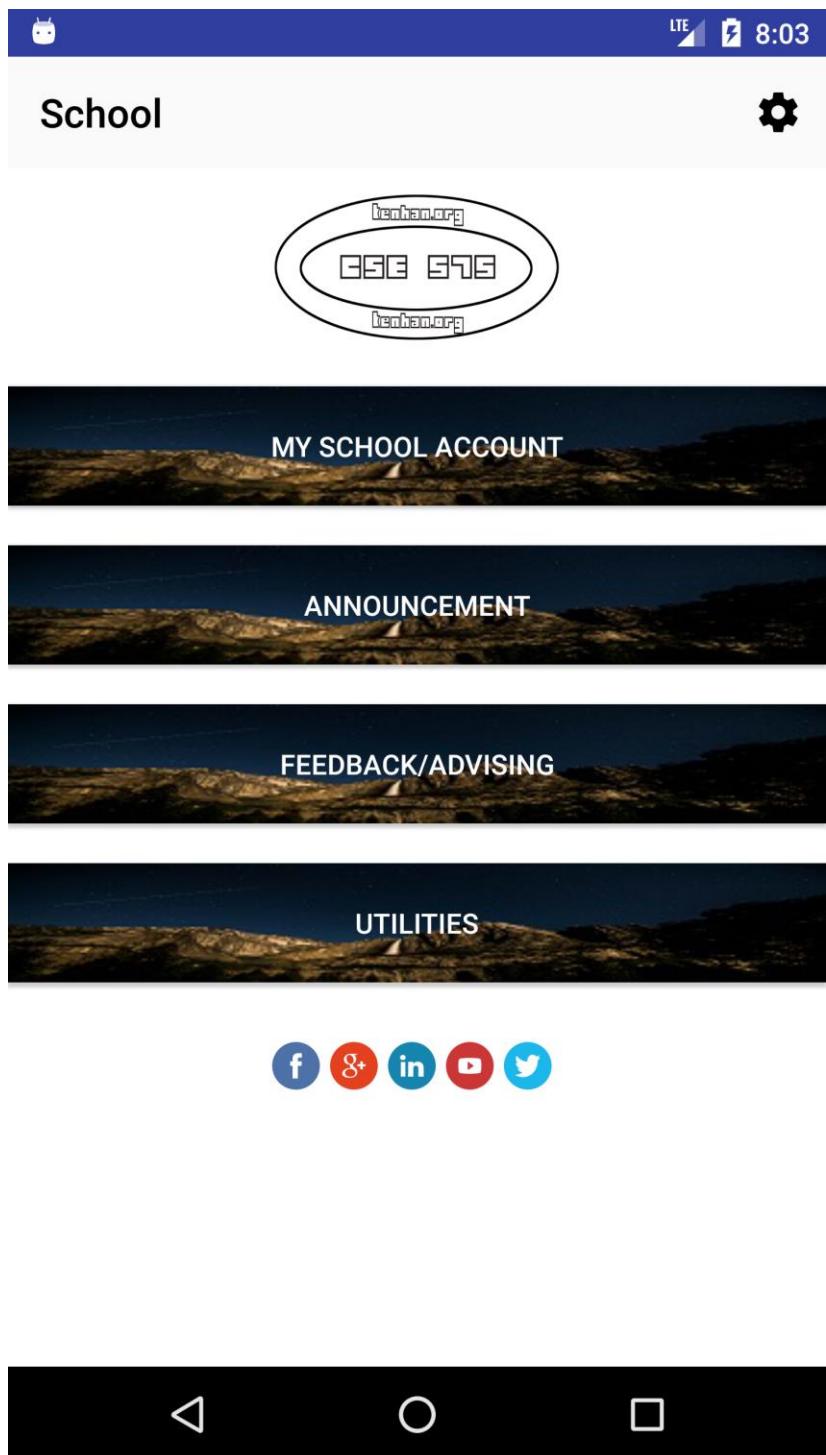
Hit the run app button to run the app on the emulator. If you run the Android Studio for the first time, you need to create a new virtual emulator that suits your need. You can customized it however you want.



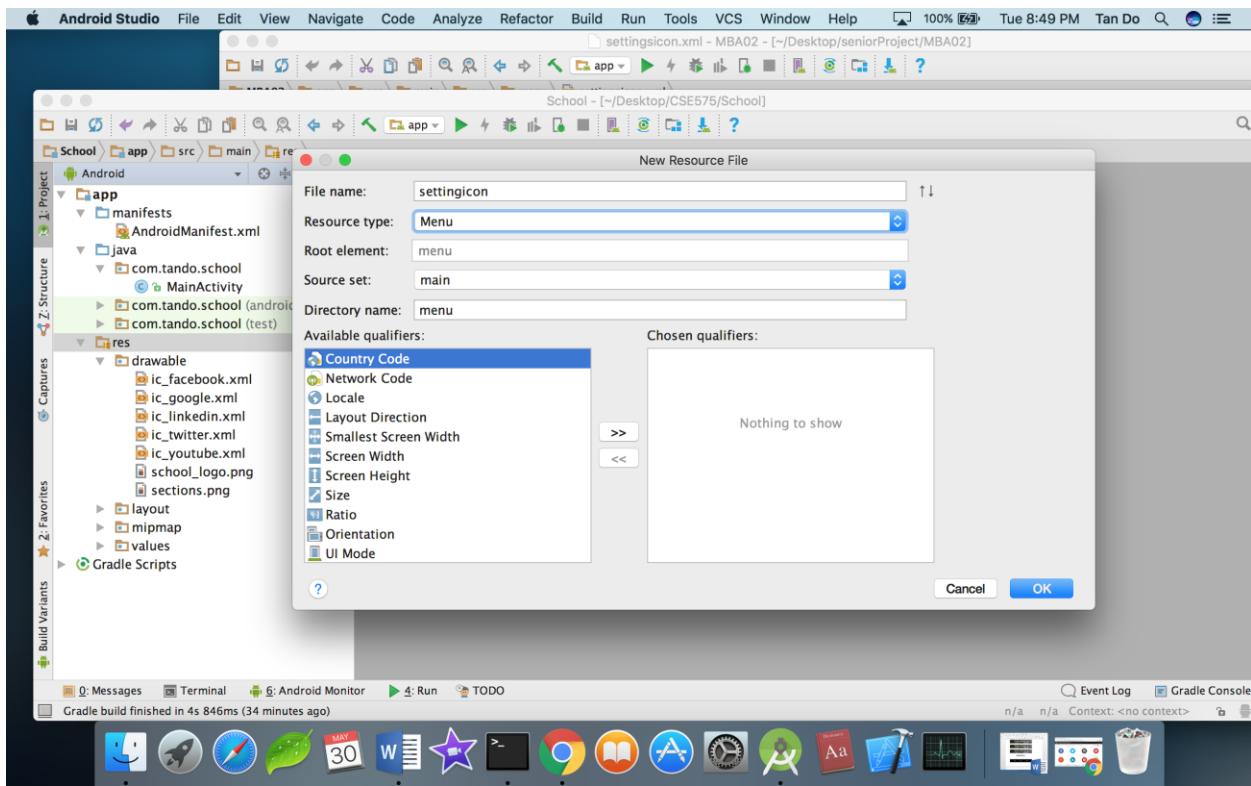
Note: Create new virtual device if you run Android Studio for the first time



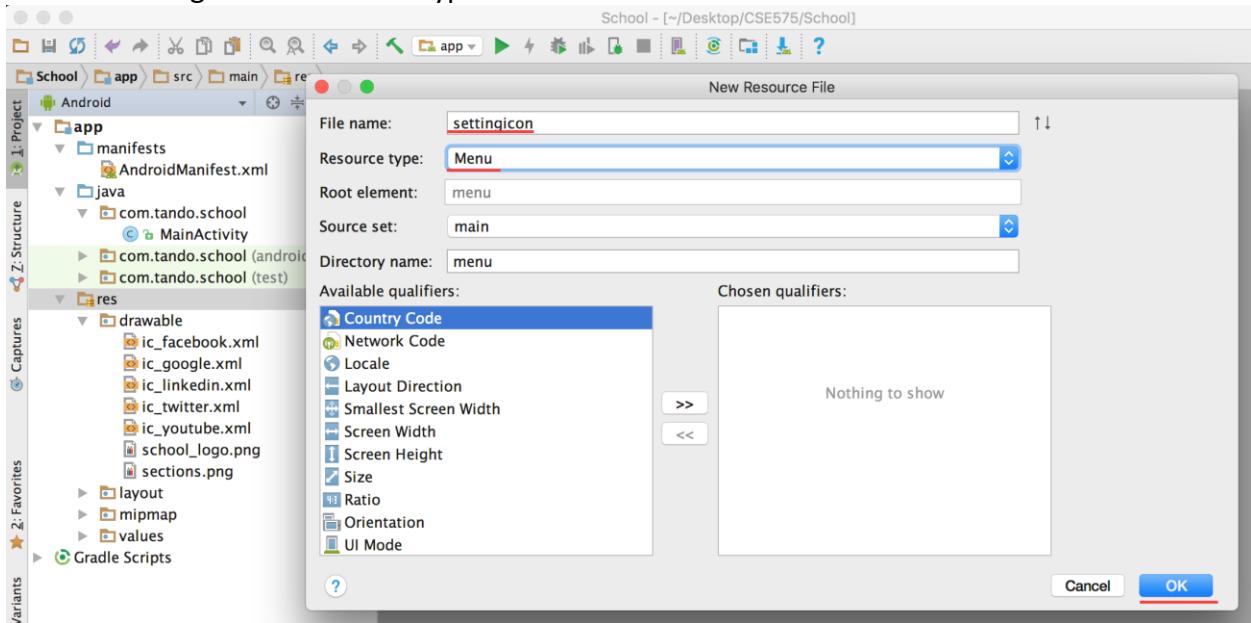
The app looked like this after I click OK. Notice that the emulator can take sometimes depends on your system's speed.



3. Create a setting or preference on the Action Bar. There will be no setting in this app so this activity is just a reference.
  - a. Right click on res directory to create new xml file



Name it settingicon > Resource Type should be Menu > OK



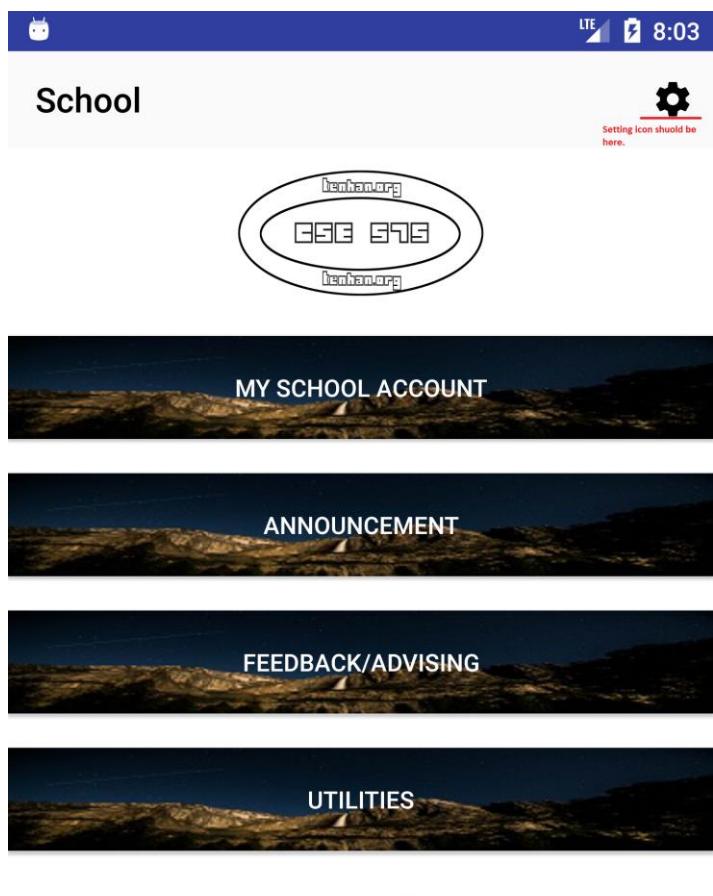
- b. Import setting icon image to drawable directory, and then click settingicon.xml. It should look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
    <item>
```

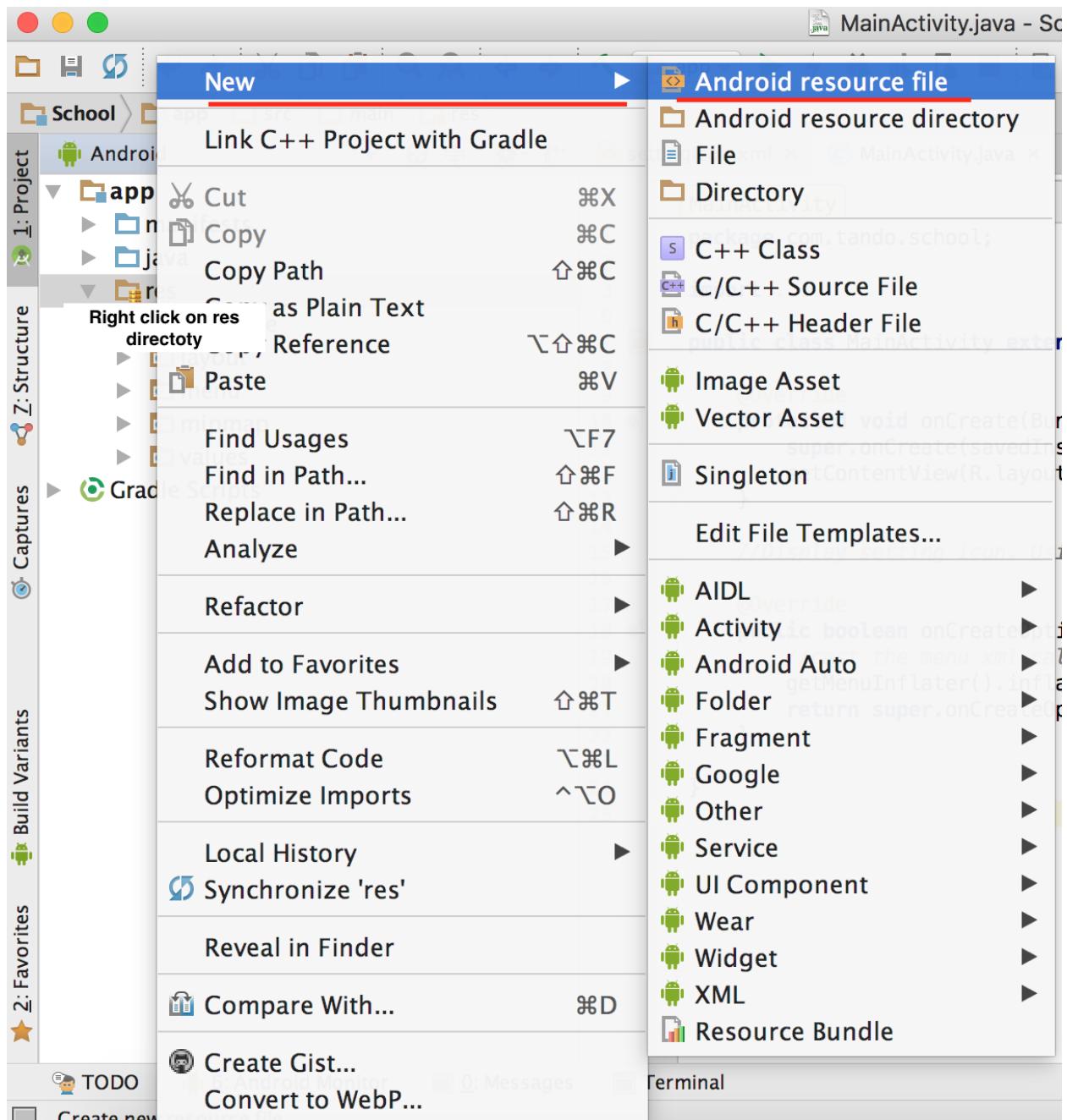
```
    android:id="@+id/settingsIcon"
    android:icon="@drawable/ic_settings"
    android:title="Setting"
    app:showAsAction="always"
/>
</menu>
c. Display the icon on the action bar. Go to Mainactivity.java > add a
onCreateOptionsMenu method
```

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    //cast the menu xml called settingIcon into the Menu inflater
    getMenuInflater().inflate(R.menu.settingIcon, menu);
    return super.onCreateOptionsMenu(menu);
}
```

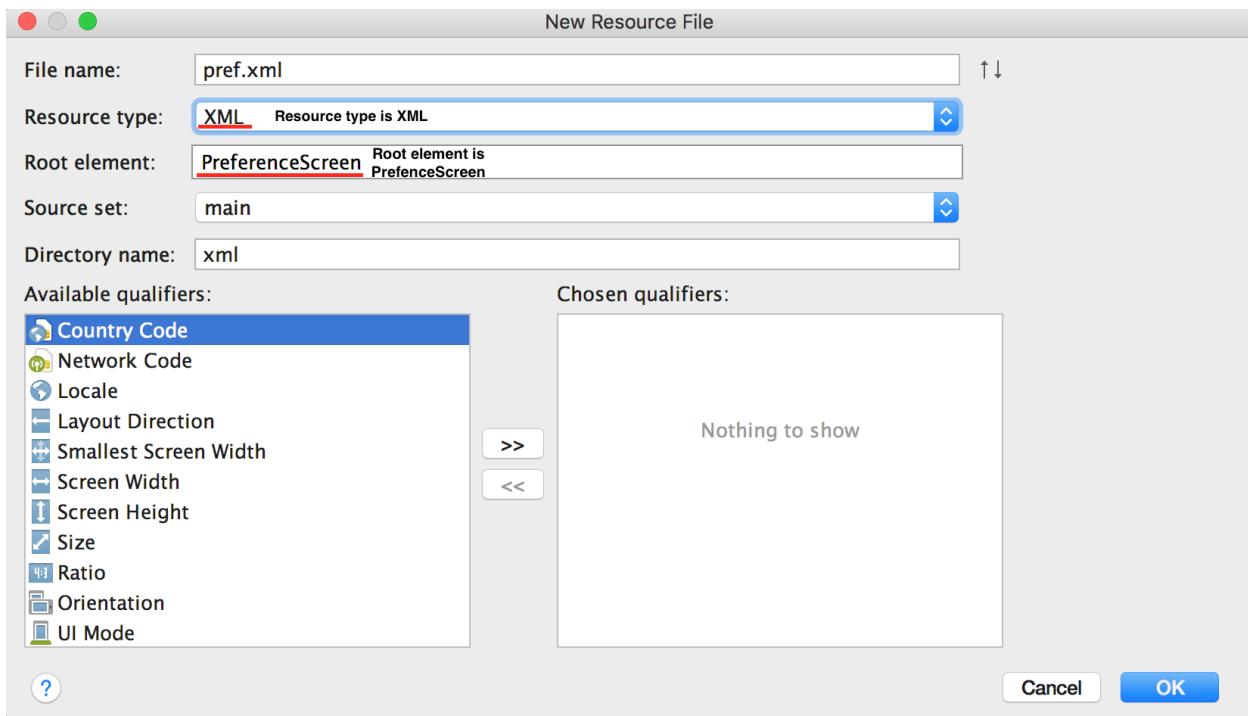
d. Click run the app to check if the setting icon appear in the actionBar



4. Creating a simple setting/preference activity when click the setting icon
  - e. Create a new xml resource file



Next



\*Note: I called pref.xml, you can call whatever you want.

Click OK

- f. Style the pref.xml. I download 2 free icons, and imported them into drawable directory as the methods I implemented earlier.

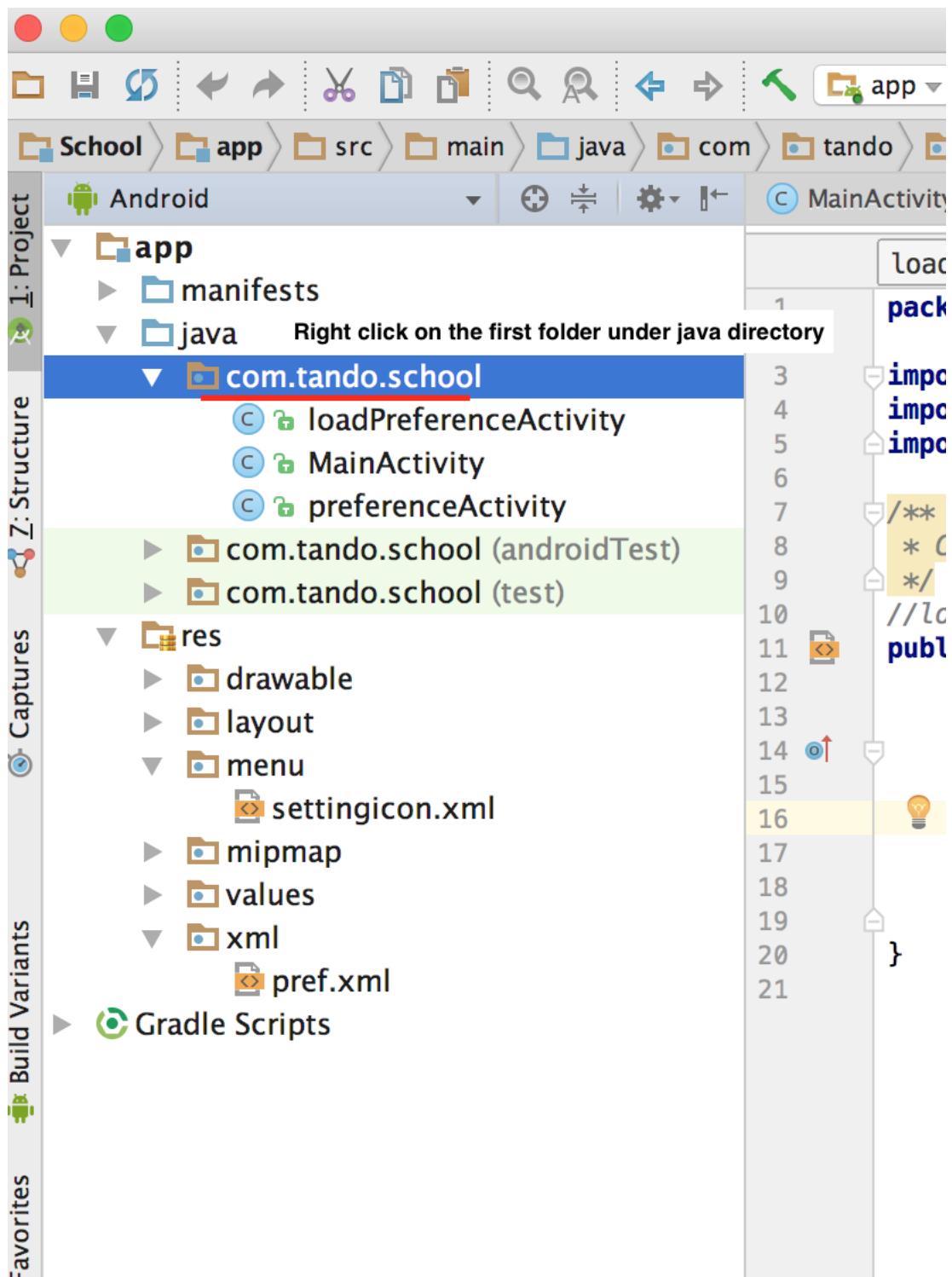
```

<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <PreferenceCategory android:title="GENERAL">
        <Preference
            android:summary="About"
            android:icon="@drawable/ic_about_24px"> *icon attribute is to display the icon
    </Preference>
</PreferenceCategory>
<PreferenceCategory android:title="SUPPORT">
    <Preference
        android:summary="Email Developers"
        android:icon="@drawable/ic_email_24px">
        <intent
            android:action="android.intent.action.VIEW"
            android:data="mailto: youremail@domain.com"> *intent is to open a different activity instead
            of a settings screen, such as other activity,
            web pages or send email.
    </Preference>
</PreferenceCategory>
</PreferenceScreen>

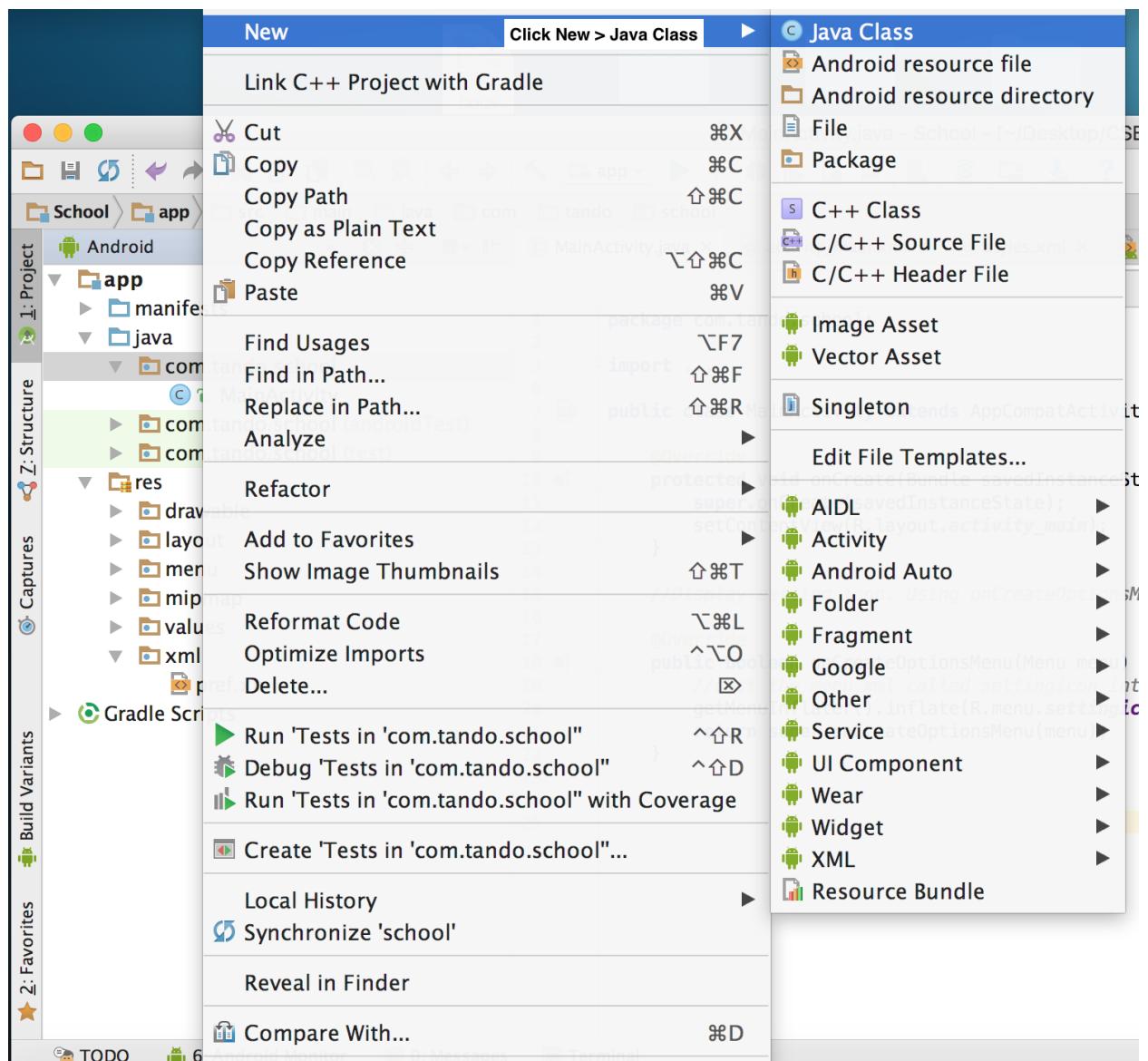
```

- g. Add java code to make the pref.xml become an activity.

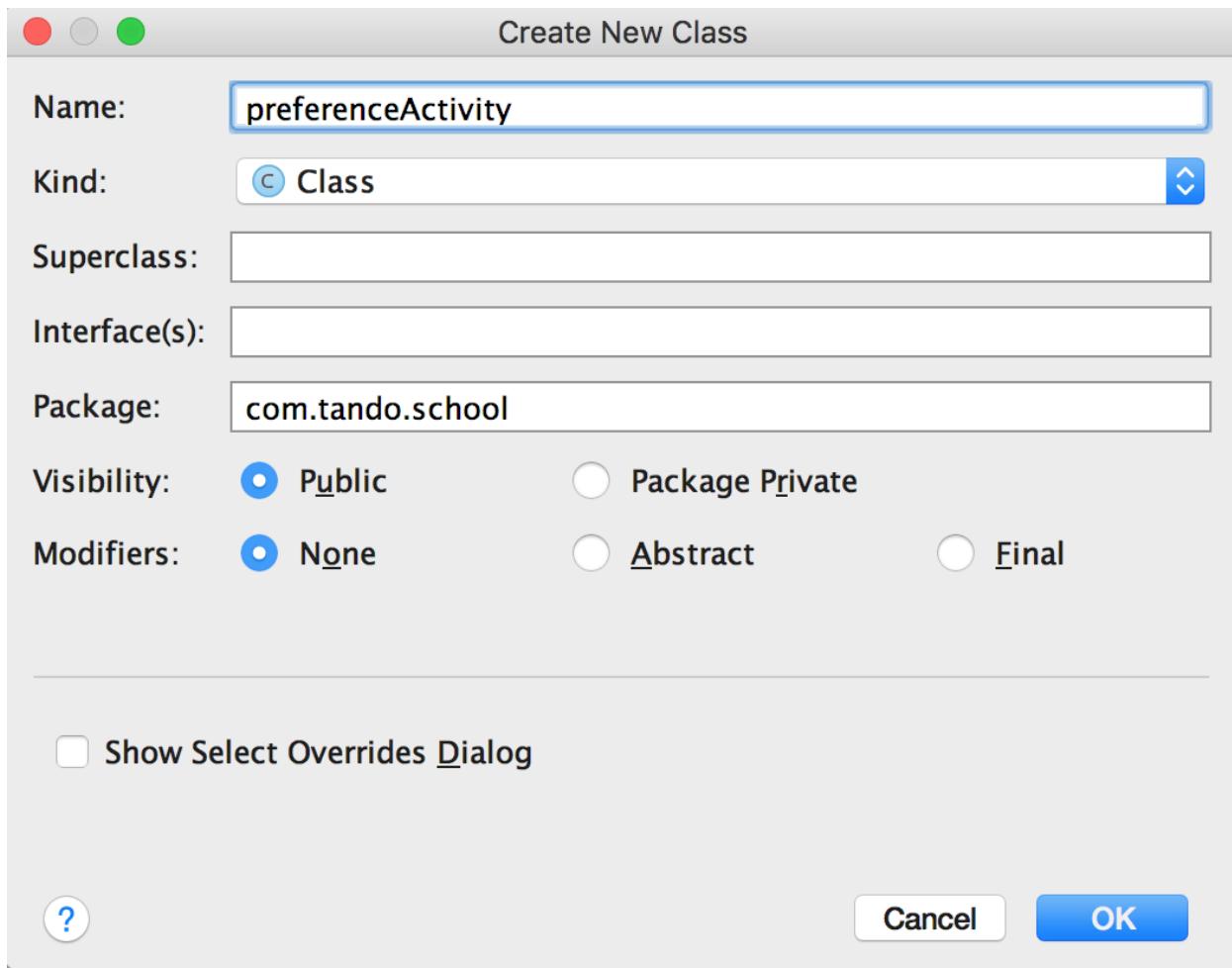
- Create new class called preferenceActivity.java



- Next



- Next



- preferenceActivity.java

```
package com.tando.school;
```

```
import android.os.Bundle;
import android.preference.PreferenceFragment;
import android.support.annotation.Nullable;

/**
 * Created by tando on 5/31/17.
 */
//use PreferenceFragment class
public class preferenceActivity extends PreferenceFragment {
    //use onCreate method to get the pref.xml
    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //use addPreferencesFromResource to add the pref.xml
        addPreferencesFromResource(R.xml.pref);
    }
}
```

- Next, create another java class called loadPreferenceActivity.java using the same method above. This class is to load our pref.xml as an activity

- loadPreferenceActivity.java  
`package com.tando.school;`

```
import android.app.Activity;
import android.os.Bundle;
import android.support.annotation.Nullable;

/**
 * Created by tando on 5/31/17.
 */
//load the pref.xml into an activity
public class loadPreferenceActivity extends Activity {
    @Override
    //use onCreate method
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //load preferenceActivity as an activity
        getFragmentManager().beginTransaction().replace((android.R.id.content),
            new preferenceActivity()).commit();
    }
}
```

- Next, go to MainActivity.java to implement the method to display the loadPreferenceActivity.java when user clicks the setting icon on actionbar
- loadPreferenceActivity.java  
`package com.tando.school;`

```
import android.app.Activity;
import android.os.Bundle;
import android.support.annotation.Nullable;

/**
 * Created by tando on 5/31/17.
 */
//load the pref.xml into an activity
public class loadPreferenceActivity extends Activity {
    @Override
    //use onCreate method
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //load preferenceActivity as an activity
        getFragmentManager().beginTransaction().replace((android.R.id.content),
            new preferenceActivity()).commit();
    }
}
```

- Next, customized the theme for loadPreferenceActivity activity
- Go to styles.xml
- Add follow lines of code:

```
<!-- Customized preference activity -->
<style name="PreferenceScreen" parent="android:Theme.Material.Light">
```

- ```
<item name="android:colorPrimary">#FFF</item>
<item name="android:colorPrimaryDark">@color/colorPrimaryDark</item>
<item name="android:colorAccent">#000</item>
<item name="android:textColorPrimary">#000</item>
</style>
```
- Next, go to manifests directory > click AndroidManifest.xml
  - Add following lines of code to have a new activity

```
<activity android:name=".loadPreferenceActivity"
    android:parentActivityName=".MainActivity"
    android:label="Preference"
    android:theme="@style/PreferenceScreen">
</activity>
```
  - Notice that parentActivityName is the back arrow on the actionbar, so basically `android:parentActivityName=".MainActivity"` means when user clicks on the back arrow, it will go back to the Mainactivity. Theme should be the theme that I customized earlier.
  - Now, click the run app button to see the result



## ← Preference

### GENERAL

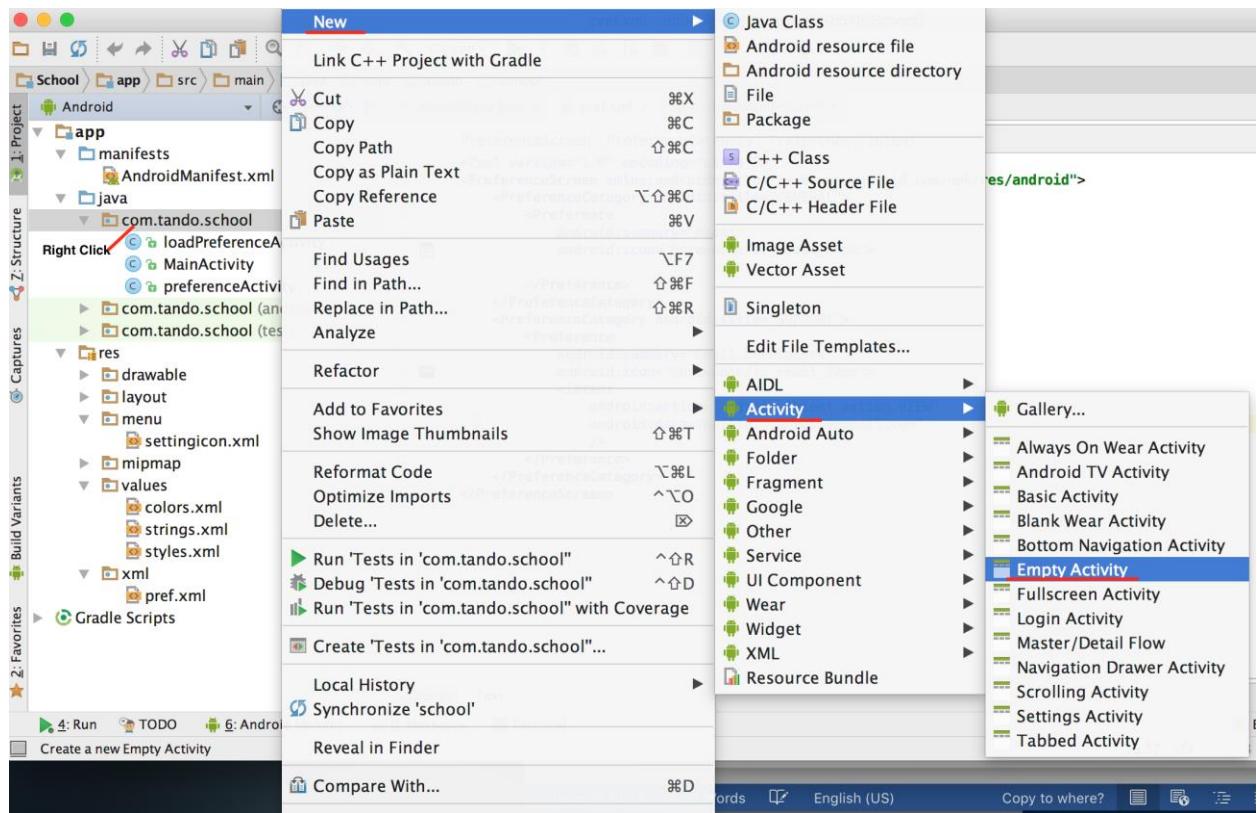
 About

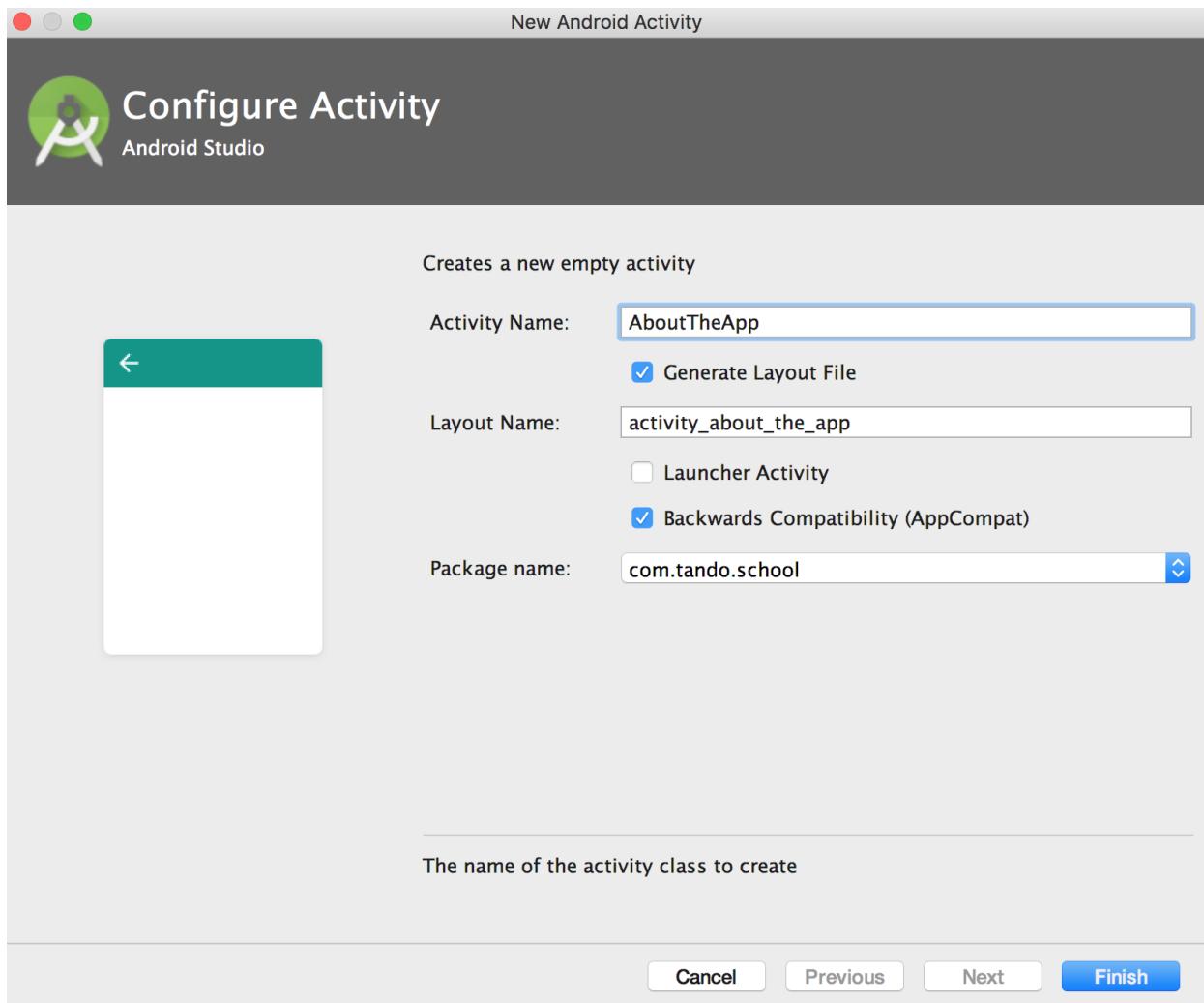
### SUPPORT

 Email Developers



- h. Create a new activity for description of the app when user clicks on the About in the preference.
- Create new empty activity named AboutTheApp





- Next open AboutTheApp.xml to style the UI of this activity

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:app="http://schemas.android.com/apk/res-auto"  
    xmlns:tools="http://schemas.android.com/tools"  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    tools:context="com.tando.school.AboutTheApp"  
    android:orientation="vertical"  
    android:background="#FFF">  
    <ScrollView  
        android:layout_width="match_parent"  
        android:layout_height="match_parent">  
        <LinearLayout  
            android:orientation="vertical"  
            android:layout_width="match_parent"  
            android:layout_height="match_parent">  
            <!--Logo -->  
            <ImageView  
                android:id="@+id/logo"
```

```

        android:layout_width="150dp"
        android:layout_height="100dp"
        android:background="@drawable/school_logo"
        android:layout_gravity="center"
        android:onClick="backHome"/>
<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="School App"
        android:textColor="#000"
        android:textStyle="bold"
        android:textSize="25dp"
        android:gravity="center"/>
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Version 1.0.0"
        android:layout_gravity="right"
        android:layout_marginRight="10dp"/>
<TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:textColor="#000"
        android:layout_margin="20dp"
        android:text="This app is developing by Tan Do. He is a student of CSUSB, and he developed this app
as a project for his internship with professor Ken Han. This app is an informative app where users (students) can
use to track, check, get notifications from the school, and also the users will be provided with some useful
utilities. "/>
</LinearLayout>
</ScrollView>
</LinearLayout>

```

- Next, go to AboutTheApp.java to make the logo functioning

```
package com.tando.school;
```

```

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;

public class AboutTheApp extends AppCompatActivity {

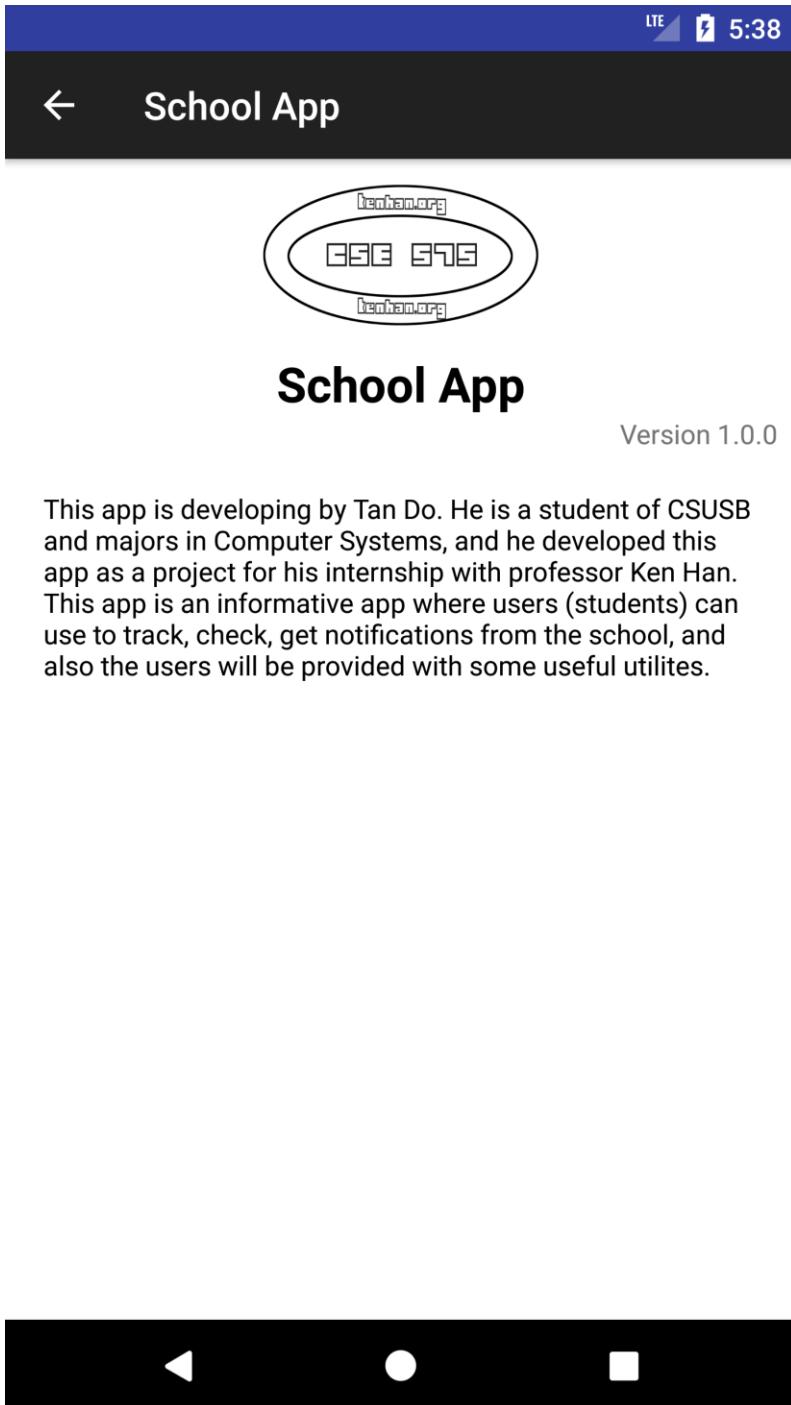
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_about_the_app);
    }
    //as I put onClick method in the ImageView in its xml file we will call it here
    public void backHome (View view) {
        //when user clicks the image (logo) it will go to the MainActivity
        startActivity(new Intent(this, MainActivity.class));
    }
}

```

- ```
    }
}

• Next, go to pref.xml to make the About icon functioning by using intent inside the Preference just like we did earlier.

<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <PreferenceCategory android:title="GENERAL">
        <Preference
            android:summary="About"
            android:icon="@drawable/ic_about_24px">
            <intent
                android:action="android.intent.action.VIEW"
                android:targetPackage="com.tando.school"
                android:targetClass="com.tando.school.AboutTheApp">
            </intent>
        </Preference>
    </PreferenceCategory>
    <PreferenceCategory android:title="SUPPORT">
        <Preference
            android:summary="Email Developers"
            android:icon="@drawable/ic_email_24px">
            <intent
                android:action="android.intent.action.VIEW"
                android:data="mailto: tantdo89@gmail.com"
                />
        </Preference>
    </PreferenceCategory>
</PreferenceScreen>
```
- Next hit the run app button to see the result



## 5. Make the social icons functioning

- First go back to the activity\_main.xml > add onClick function for all of the social media icons.
- Note\*: name the functions something that meaningful.
- Example:

```

74
75
76
77
78      f
79
80         android:layout_width="wrap_content"
81         android:layout_height="wrap_content"
82         android:layout_weight="1"
83         android:src="@drawable/ic_facebook"
84         android:paddingRight="5dp"
85         android:onClick="fbClick"/>
86
87      g+
88
89         android:id="@+id/googlePlus"
90         android:layout_width="wrap_content"
91         android:layout_height="wrap_content"
92         android:layout_weight="1"
93         android:src="@drawable/ic_google"
94         android:paddingRight="5dp"
95         android:onClick="googleClick"/>
96
97      in
98
99

```

onClick function

- Next, go to MainActivity.java to add following codes:
- Note\*: the url of the social media will be put inside the quotation marks

```

//link to FaceBook
public void fbClick (View view) {
    //use startActivity and Intent methods to parse the link
    startActivity(new Intent(Intent.ACTION_VIEW,
    Uri.parse("https://www.facebook.com/flitetest/")));
}
//google+
public void googleClick (View view) {
    //use startActivity and Intent methods to parse the link
    startActivity(new Intent(Intent.ACTION_VIEW,
    Uri.parse("https://plus.google.com/u/0/+google")));
}
//LinkedIn
public void linkedInClick (View view) {
    //use startActivity and Intent methods to parse the link
    startActivity(new Intent(Intent.ACTION_VIEW,
    Uri.parse("https://www.linkedin.com/in/mbacsusb/")));
}
//youtube
public void youTubeClick (View view) {
    //use startActivity and Intent methods to parse the link
    startActivity(new Intent(Intent.ACTION_VIEW,
    Uri.parse("https://www.youtube.com/channel/UCwqJShoifP6032UXYe-sD-A")));
}
//Twitter
public void twitterClick (View view) {
    //use startActivity and Intent methods to parse the link

```

```
    startActivity(new Intent(Intent.ACTION_VIEW,  
Uri.parse("https://twitter.com/Google")));  
}
```